

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Integrasi Aplikasi Web dan Android

Percobaan yang dilakukan oleh Jos Forman Tompoh (Tompoh et al., 2016) memanfaatkan *framework cordova* dan *framework ionic* sehingga dapat diakses melalui web dan android secara server lokal dengan menggunakan XAMPP. Dengan menggunakan *web-services client*, user android dan *client* admin pada web dapat berintegrasi dalam sistem pemesanan.

Pada Aplikasi yang dirancang oleh Wismarini dan Prihandono (Wismarini & Prihandono, 2020), melakukan integrasi aplikasi android dengan web services menggunakan library *volley* yang memanfaatkan REST Api sebagai perantara dengan web service nya. Sistem yang diterapkan juga masih menggunakan web server secara lokal.

2.1.2 Penerapan Metode Geocode pada Aplikasi Android

Dalam Aplikasi yang dikembangkan oleh Sandheep S, Harry John, Harikumar A dan Vinitha Panicker J (Sandheep et al., 2017), diterapkan metode *Reverse Geocoding* untuk mengubah data koordinat yang dikoleksi dengan metode *Crowdsourcing* menjadi lokasi *bus stop* terdekat pada Google Map berdasarkan koordinat yang didapat oleh user setelah turun atau naik bus. Sistem ini akan berkembang lebih cepat jika semakin banyak

user yang menggunakan, dikarenakan aplikasi akan menerima lebih banyak data dari *Crowdsourcing* dan akan lebih efektif.

Pada perancangan dan pembuatan aplikasi *Emergency Search* yang dirancang oleh Anisha Ginjala (Ginjala, 2015), Menggunakan Twitter API untuk mendapatkan postingan atau *tweet* tentang keadaan darurat, lalu ditampilkan pada *Google Maps* melalui aplikasi ini.

User juga dapat memposting atau menuliskan suatu keadaan darurat melalui aplikasi ini dan menuliskan lokasi kejadian, jika user tidak menuliskan lokasi maka secara otomatis lokasi akan tertulis menggunakan metode *Reverse Geocoding* dengan mengubah titik *Latitude* dan *Longitude* menjadi lokasi suatu tempat sehingga dapat dibaca oleh user lain nya.

2.2 Landasan Teori

2.1.1 Peta

Peta merupakan penyajian grafis dari permukaan bumi dalam skala tertentu dan digambarkan pada bidang datar melalui sistem proyeksi peta dengan menggunakan symbol-simbol tertentu sebagai perwakilan dari objek-objek spasial yang berada di permukaan bumi seperti gunung, jalan, hutan dll[2].

2.1.2 Google Maps

Google Maps adalah peta digital yang dapat digunakan untuk melihat suatu daerah[2] yang dikembangkan oleh Google dan diluncurkan secara publik pada bulan Februari tahun 2005, bermula dari program *desktop* dua orang bersaudara dari Denmark yaitu Lars Rasmussen yang ingin menyaingi program digital mapping yang sudah ada seperti MapQuest, lalu Google mendukung startup dari Rasmussens bersaudara di tahun 2004[3].

2.1.3 Google Maps API

Google Maps API adalah suatu library berbentuk JavaScript [2] yang disediakan oleh Google melalui *google play services library* yang dapat didownload secara eksternal dari situs android developer atau dari *android SDK manager*. Sehingga aplikasi android dapat mengintegrasikan fitur-fitur Google Maps untuk dimanfaatkan fungsi-fungsinya untuk digunakan pada berbagai bidang [4].

2.1.4 *Global Positioning System (GPS)*

Fitur yang sering digunakan oleh masyarakat dalam penggunaan *Google Maps* saat ini adalah GPS atau dapat dikenal dengan *Global Positioning System*, GPS merupakan sistem navigasi dan penentu lokasi berbasis satelit dengan tingkat ketelitian tinggi [5]. GPS telah dikembangkan dalam bentuk smartphone sehingga penggunaannya lebih mudah sehingga datanya dapat digunakan untuk mengambil data koordinat dari masing-masing pengguna smartphone [5].

2.1.5 *Location-Based Services (LBS)*

LBS adalah layanan berbasis lokasi, yaitu sebuah layanan berbasis internet yang berfungsi untuk mencari lokasi dengan berbasis GPS. Map dan layanan berbasis lokasi menggunakan lintang bujur bumi (*longitude* dan *latitude*). Android telah menyediakan *geocoder* yang dapat mendukung *forward* dan *reverse geocoding*[6].

Dengan *geocode* nilai lintang bujur (*longitude* dan *latitude*) dapat dikonversikan menjadi alamat yang dapat dikenali secara *plain text*[6].

2.1.6 *Geocoding*

Geocoding adalah proses konversi deskripsi lokasi berbasis *text-based* menjadi sebuah nilai koordinat. Namun proses *geocoding* ini masih memiliki kesalahan spasial yang berpengaruh terhadap

output sehingga dapat berpengaruh dalam kevalidasi dan keakurasian pengukuran lokasi dan jarak pada *geocoding* itu sendiri[7].

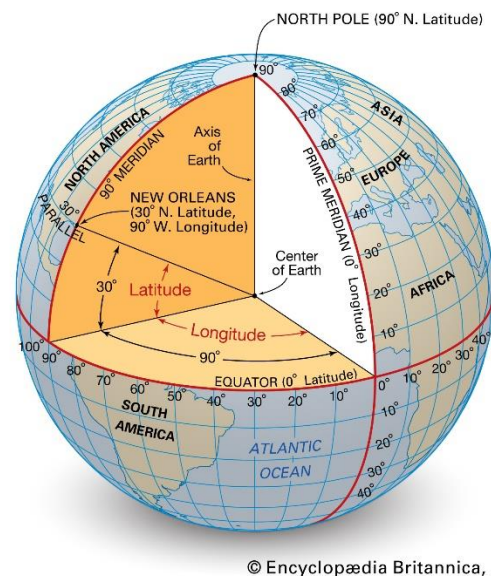
Geocoding dapat dimanfaatkan sebagai simple data analysis sampai keperluan bisnis dan manajemen kostumer[8].

Geocoding memiliki dua metode yaitu *Forward Geocoding* dan *Reverse Geocoding*. *Forward Geocoding* adalah proses konversi suatu alamat dari *plain text* menjadi koordinat geografik[9]. Sedangkan *Reverse Geocoding* adalah proses konversi koordinat geografik menjadi alamat secara *plain text* yang dapat dibaca oleh manusia[10].

Dalam penelitian ini, penulis akan menggunakan metode *Reverse Geocoding* yang mengkonversi koordinat *latitude* dan *longitude* melalui aplikasi dan dikonversi menjadi sebuah informasi berupa *plain text* yang dapat dibaca oleh user berupa nomor meja. Sehingga secara otomatis jika user duduk di salah satu lokasi koordinat yang sudah di tetapkan, maka secara otomatis aplikasi mengambil lokasi user saat ini dan menjalankan proses *Reverse Geocoding* dan mengubah lokasi koordinat menjadi sebuah informasi berupa nomor meja yang ditempati.

2.1.7 Longitude dan Latitude

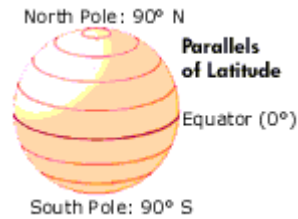
Dalam penentuan koordinat lokasi di GPS sendiri ditentukan oleh *longitude*, *latitude*. *Latitude* dan *longitude* dihitung berdasarkan sudut yang bertumpu dari inti bumi[11].



Gambar 1.1 Ilustrasi inti bumi terhadap longitude dan latitude

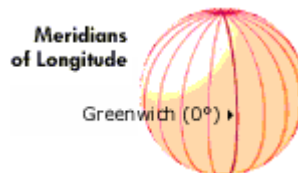
Latitude adalah garis yang melintang secara horizontal sejajar dengan garis *equator* yang memutar bumi dan berlawanan dengan garis meridian utama. Dengan batas + dan - berdasarkan jauh dekat nya garis dengan garis *equator* bumi. Dari kutub utara ke *equator* adalah garis melintang positif (+) dan garis melintang setelah *equator* sampai ke kutub selatan adalah garis melintang negatif (-)[12].

Garis *equator* adalah titik pusat dari *latitude* sehingga garis *equator* memiliki 0 derajat latitude. Sudut *latitude* akan lebih besar semakin garis *latitude* menjauhi garis pusat *equator* sehingga kutub utara dan kutub selatan memiliki sudut *latitude* sebesar 90 derajat.

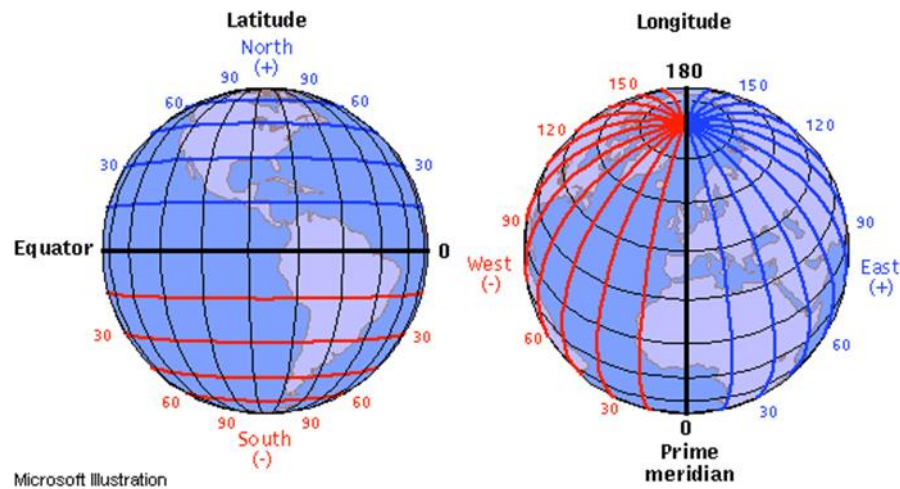


Gambar 1.2 Ilustrasi garis melintang *latitude* terhadap garis *equator*

Longitude juga disebut sebagai *meridian* [13] adalah garis melintang secara vertikal yang sejajar dengan garis meridian utama dan berlawanan dengan garis *equator* bumi. Dengan batas + dan – berdasarkan jauh dekat nya dengan garis meridian utama. Wilayah Greenwich di Inggris adalah titik pusat dari *longitude* atau memiliki *longitude* 0 derajat[14] sehingga perhitungan dihitung dari arah wilayah Greenwich ke kanan adalah kutub positif (+) dari *longitude* dan dari wilayah Greenwich kekiri adalah kutub negatif (-) dari *longitude*[12].



Gambar 1.3 Ilustrasi garis vertikal *longitude* terhadap garis meridian utama



Gambar 1.4 Ilustrasi perbandingan dari garis *latitude* dan *longitude*

2.1.8 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang dapat mencakup sistem operasi, middleware dan aplikasi[15].

Karena android berbasis linux maka android termasuk *open source* sehingga siapapun dapat memodifikasi sistem operasi nya secara bebas, berbeda dengan iOS yang bersifat closed source[16]. Bahasa pemrograman yang digunakan pada android adalah Java[17].

2.1.9 Aplikasi

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari user atau pengguna aplikasi tersebut dengan tujuan untuk mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut.

Secara umum, pengertian aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai dengan kemampuan yang dimilikinya, aplikasi merupakan suatu perangkat komputer yang siap pakai bagi user[6].

2.1.10 Bahasa Pemrograman PHP

PHP atau dapat dikenal sebagai *PHP Hypertext Processor* digunakan sebagai bahasa script dalam pengembangan web yang dimasukan pada dokumen HTML. Penggunaan PHP memungkinkan web berkomunikasi dengan aplikasi desktop maupun android menggunakan fungsi API (*Application Programming Interface*) sehingga aplikasi dapat menyimpan sebuah database pada MySQL secara online.

2.1.11 Database MySQL

MySQL adalah sistem database open source yang paling populer di seluruh dunia dikarenakan MySQL adalah open source sehingga semua orang dapat menggunakan dan mengembangkan MySQL untuk segala kebutuhan secara gratis.

MySQL adalah sistem database yang paling sering digunakan aplikasi berbasis web[2] dan dapat berkomunikasi satu sama lain dengan aplikasi dengan perantara API.

2.1.12 Unified Modeling Language (UML)

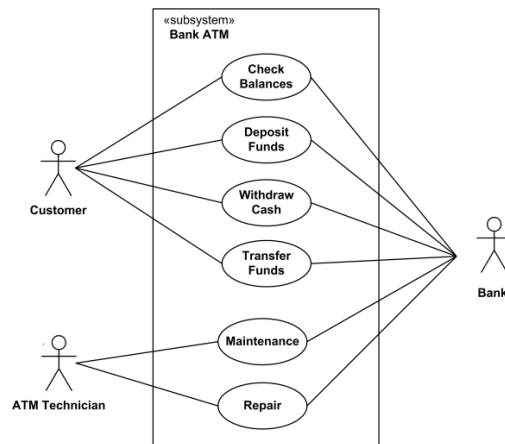
Unified Modeling Language atau yang dapat disebut UML adalah satu metode pemodelan visual yang digunakan untuk perancangan dan pembuatan sebuah aplikasi atau *software* yang berorientasikan pada objek atau *Object Oriented Software*. [18]

UML merupakan sebuah standar penulisan yang memiliki rancangan-rancangan yang berisi bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang lebih spesifik dan detail.[18]

UML memiliki beberapa diagram yang sering digunakan dalam pengembangan sebuah sistem, yaitu :

1. *Use Case Diagram*

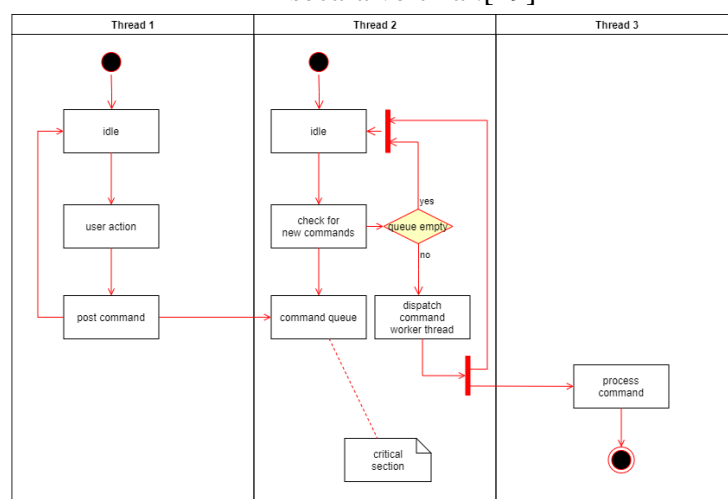
- a. suatu sistem *Use Case Diagram* adalah salah satu jenis dari diagram UML yang menggambarkan hubungan interaksi antara sistem dan aktor(pengguna). *Use Case Diagram* dapat mendeskripsikan tipe-tipe interaksi antara pengguna sistem dengan sistem yang digunakan.[19]



Gambar 1.5 Contoh Use Case Diagram

2. Activity diagram

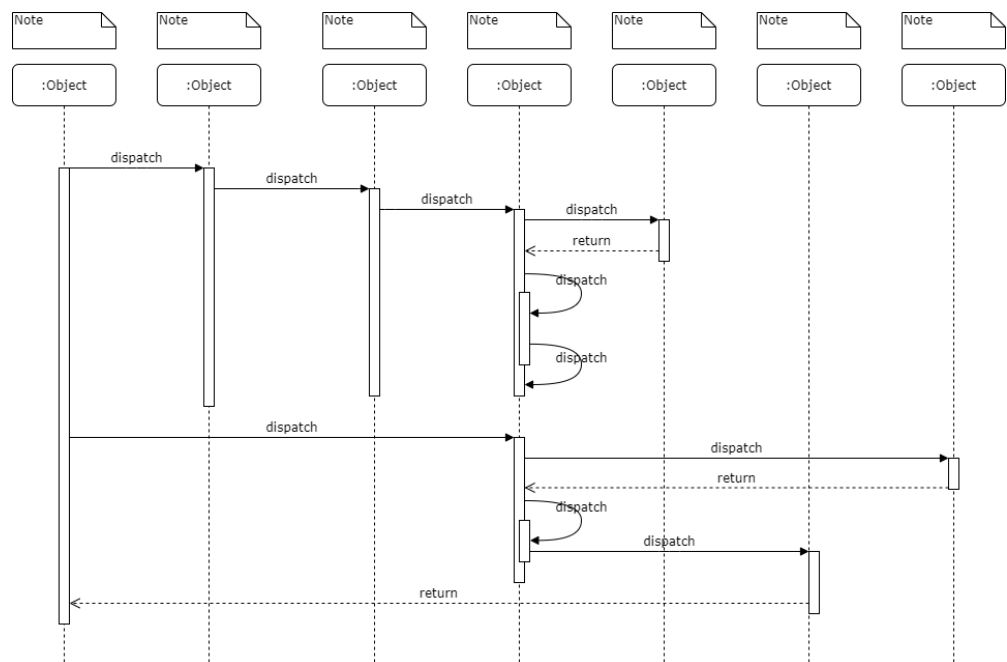
- a. merupakan sebuah diagram yang dapat menggambarkan model berbagai proses yang terjadi pada sistem. Seperti runtutan proses berjalannya suatu sistem dan digambarkan secara vertikal.[19]



Gambar 1.6 Contoh Activity Diagram

3. Sequence Diagram

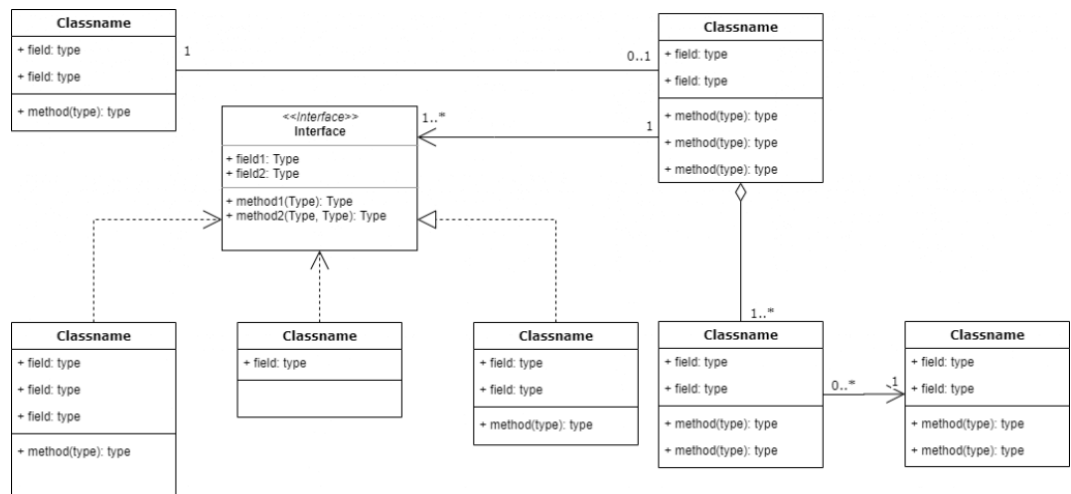
- a. *Sequence Diagram* menggambarkan Interaksi antar objek didalam dan di sekitar sistem yang berupa pesan yang digambarkan terhadap waktu.[18]



Gambar 1.7 Contoh Sequence Diagram

4. Class Diagram

- a. *Class Diagram* merupakan gambaran struktur dan deskripsi dari suatu class, package dan objek yang saling berhubungan seperti diantaranya pewarisan, asosiasi dan lainnya.[18]



Gambar 1.8 Contoh Class Diagram

2.1.13 JavaScript Object Notation (JSON)

JSON adalah format pertukaran data yang ringan dan dapat lebih mudah dimengerti. JSON juga lebih mudah diproses oleh mesin atau software[20].

2.1.14 Application Programming Interface (API)

API adalah software perantara yang berfungsi untuk melakukan komunikasi antar dua aplikasi[21]. Contoh nya adalah aplikasi *mobile* mengirimkan perintah input yang digunakan untuk menampilkan data dari database dikirimkan melalui API, lalu API meneruskan permintaan input tersebut ke Database, lalu dari Database mengirimkan kembali output yang diminta oleh aplikasi *mobile* melalui API berupa *JSON text* dan ditampilkan oleh aplikasi *mobile*.

2.1.15 REST API

REST API adalah salah satu teknologi API yang sering digunakan ketika akan mengembangkan sebuah WEB API. REST API sendiri memiliki sifat *stateless*, yaitu dimana setiap kali *request* harus menyertakan semua data dan parameter dengan lengkap dan benar ketika mengakses suatu *endpoint*. [22]

Pada Arsitektur REST, REST server menyediakan *resources* dan REST client akan mengakses dan menampilkan *resource* tersebut untuk pengguna selanjutnya. *Resource* direpresentasikan dalam bentuk format teks, seperti JSON atau XML. [22]

REST memiliki standarisasi dalam pemakaian yaitu URL dan HTTP *method*. HTTP *method* digunakan untuk mengetahui fungsi dari URL yang diakses, sehingga mempermudah dalam penulisan URL. [22]

Jenis HTTP *method* yang sering digunakan adalah :

1. **GET**, digunakan untuk membaca sebuah *record* atau data.
2. **POST**, digunakan untuk menambahkan *record* atau data.
3. **PUT**, digunakan untuk mengubah semua *field* dalam sebuah *record* atau data.
4. **PATCH**, digunakan untuk mengubah beberapa *field* dalam sebuah *record* atau data.

5. **DELETE**, digunakan untuk menghapus sebuah *record* atau data.

2.1.16 Volley

Volley adalah HTTP *library* pada *android studio* [23] untuk mempermudah dan kecepatan proses koneksi aplikasi android dengan jaringan.[1]

Pada penelitian ini penulis menggunakan *library Volley* untuk menjalin komunikasi antara aplikasi android dengan *REST API* yang digunakan.

Volley digunakan untuk mengirim input dari aplikasi menuju *REST API* yang akan diteruskan untuk membuat data atau mengambil data pada database.

Data yang dikirimkan oleh Volley ke *REST API* berupa parameter yang dibutuhkan yang nantinya akan di olah di *Function* yang berada di *REST API*.

Volley akan menerima output yang dikirim kan dari *REST API* berupa format teks JSON.

2.1.17 Entity Relationship Diagram (ERD)

ERD adalah suatu diagram untuk menggambarkan desain konseptual dari model konseptual suatu basis data relasional[24]. ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi satu sama lain[2].

ERD memiliki hubungan erat dengan *Data Flow Diagram* (DFD) untuk menampilkan sebuah data yang disimpan. Yang bertujuan untuk memvisualisasikan proses data yang dapat saling terhubung dan dapat konstruksi data relasional[25]. ERD memiliki beberapa komponen yang dipakai, yaitu :

1. Entitas

Entitas merupakan kumpulan dari objek-objek yang dapat teridentifikasi secara Unik. Di dalam ERD, entitas dilambangkan dengan persegi panjang. Dan entitas lemah akan digambarkan dengan bentuk persegi panjang kecil didalam persegi panjang yang besar[25].

2. Atribut

Setiap entitas memiliki bermacam macam atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Penggunaan atribut kunci atau *key* adalah pembeda dari entitas dan atribut yang diwakili dengan simbol elips[25].

Macam-macam atribut adalah sebagai berikut :

a. Atribut Kunci

Atribut kunci adalah atribut yang digunakan untuk menentukan data yang bersifat unik. Biasanya data dari atribut *key*

berbentuk angka seperti NIK, NIM ,dan lain sebagainya[25].

b. Atribut Simpel

Atribut simpel adalah atribut yang tidak dapat dipecah lagi dan bernilai tunggal. Seperti alamat rumah, kantor, tahun terbit jurnal dan lain sebagainya[25].

c. Atribut Multinilai (*Multivalue*)

Atribut multi nilai adalah atribut yang memiliki sekelompok nilai untuk tiap entitasnya. Seperti kumpulan namaa pengarang dalam novel[25].

d. Atribut Gabungan (*Composite*)

Atribut gabungan adalah atribut yang berasal dari susunan atribut yang lebih kecil pada artian tertentu. Seperti data yang berhubungan pada nama lengkap yaitu nama depan, tengah dan nama belakang[25].

e. Atribut Derivatif

Atribut Derivatif adalah atribut yang berasal dari atribut lain yang tidak bersifat wajib ditulis pada ERD. Seperti

Usia, selisih waktu, kelas dan lain sebagainya[25].

3. Relasi

Relasi adalah sebuah hubungan antara beberapa jenis entitas yang berasal dari himpunan entitas-entitas yang berbeda. Relasi ini dilambangkan dengan bentuk ketupat[25]. Dalam ERD relasi yang digunakan ada tiga, yaitu :

a. One to One

One to one berarti setiap entitas hanya boleh memiliki relasi dengan satu entitas yang lain. Seperti mahasiswa dengan data NIM, satu mahasiswa hanya memiliki satu NIM[25].

b. One to Many

One to many adalah hubungan antara satu entitas dengan beberapa entitas dan sebaliknya. Seperti sekolah dengan siswa, Sekolah memiliki beberapa siswa, dan siswa hanya memiliki satu sekolah[25].

c. Many to Many

Many to many adalah hubungan antara beberapa entitas yang memiliki lebih dari suatu relasi. Seperti kelas memiliki beberapa stop kontak, dan beberapa stop kontak dimiliki beberapa ruangan[25].

4. Garis

Garis berfungsi untuk menghubungkan antar atribut sebagai bentuk hubungan entitas dari diagram ERD itu sendiri[25].

DAFTAR PUSTAKA

- [1] T. D. Wismarini and A. Prihandono, "RANCANG BANGUN APLIKASI ANDROID TERINTEGRASI WEB SERVICE DENGAN VOLLEY UNTUK LAYANAN PUBLIK," *Dinamik*, vol. 25, no. 1, pp. 10–19, Jun. 2020, doi: 10.35315/dinamik.v25i1.7515.
- [2] R. Ariyanti and I. Kanedi, "PEMANFAATAN GOOGLE MAPS API PADA SISTEM INFORMASI GEOGRAFIS DIREKTORI PERGURUAN TINGGI DI KOTA BENGKULU," vol. 11, no. 2, p. 11, 2015.
- [3] S. McQuire, "One map to rule them all? Google Maps as digital technical object," *Commun. Public*, vol. 4, no. 2, pp. 150–165, Jun. 2019, doi: 10.1177/2057047319850192.
- [4] P. Doshi, P. Jain, and A. Shakwala, "Location Based Services and Integration of Google Maps in Android," *Int. J. Eng. Comput. Sci.*, vol. 3, no. 03, Art. no. 03, Mar. 2014, Accessed: Jan. 22, 2021. [Online]. Available: <http://103.53.42.157/index.php/ijecs/article/view/190>
- [5] A. Muawwal, "Implementasi Teknologi GPS Tracking Smartphone Sebagai Aplikasi Monitoring Lokasi Anak," p. 5.
- [6] S. Alfeno and R. E. C. Devi, "Implementasi Global Positioning System (GPS) dan Location Based Service (LSB) pada Sistem Informasi Kereta Api untuk Wilayah Jabodetabe," *J. SISFOTEK Glob.*, vol. 7, no. 2, Art. no. 2, Sep. 2017, doi: 10.38101/sisfotek.v7i2.146.
- [7] Z. Yin, A. Ma, and D. W. Goldberg, "A deep learning approach for rooftop geocoding," *Trans. GIS*, vol. 23, no. 3, pp. 495–514, 2019, doi: <https://doi.org/10.1111/tgis.12536>.
- [8] "What is geocoding?—ArcMap | Documentation." <https://desktop.arcgis.com/en/arcmap/latest/manage-data/geocoding/what-is-geocoding.htm> (accessed Jan. 22, 2021).
- [9] L. Zeigermann, "Opencagegeo: Stata Module for Forward and Reverse Geocoding," p. 10.
- [10] "Overview | Geocoding API," *Google Developers*. <https://developers.google.com/maps/documentation/geocoding/overview> (accessed Jan. 22, 2021).
- [11] "GSP 270: Latitude and Longitude." http://gsp.humboldt.edu/OLM/Lessons/GIS/01%20SphericalCoordinates/Latitude_and_Longitude.html (accessed Dec. 17, 2020).
- [12] "Understanding Latitude and Longitude." <https://journeynorth.org/tm/LongitudeIntro.html> (accessed Jan. 22, 2021).
- [13] C.-12 Foundation, "Longitude | CK-12 Foundation." <https://www.ck12.org/book/physics-from-stargazers-to-starships/section/10.2/> (accessed Jan. 22, 2021).
- [14] "Greenwich meridian | geography," *Encyclopedia Britannica*. <https://www.britannica.com/place/Greenwich-meridian> (accessed Jan. 22, 2021).

- [15] J. F. Tompoh, S. R. Sentinuwo, and A. A. E. Sinsuw, "Rancang Bangun Aplikasi Pemesanan Menu Makanan Restoran Berbasis Android," *J. Tek. Inform.*, vol. 9, no. 1, Art. no. 1, Oct. 2016, doi: 10.35793/jti.9.1.2016.13749.
- [16] "Android Definition." <https://techterms.com/definition/android> (accessed Jan. 22, 2021).
- [17] "What is android - javatpoint," www.javatpoint.com. <https://www.javatpoint.com/android-what-where-and-why> (accessed Jan. 22, 2021).
- [18] M. T. Prihandoyo, "Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis Web," *J. Inform.*, p. 4, 2018.
- [19] D. Intern, "Apa itu UML? Beserta Pengertian dan Contohnya," *Dicoding Blog*, May 11, 2021. <https://www.dicoding.com/blog/apa-itu-uml/> (accessed Dec. 24, 2021).
- [20] "JSON." <https://www.json.org/json-en.html> (accessed Dec. 24, 2021).
- [21] "What is an API? (Application Programming Interface)," *MuleSoft*. <https://www.mulesoft.com/resources/api/what-is-an-api> (accessed Dec. 24, 2021).
- [22] D. S. Agnes, "Memahami API, REST API, dan RESTful API," *wripolinema*, Sep. 17, 2020. <https://medium.com/wripolinema/memahami-api-rest-api-dan-restful-api-5fd2327edd3c> (accessed Dec. 24, 2021).
- [23] "Volley overview," *Android Developers*. <https://developer.android.com/training/volley> (accessed Dec. 24, 2021).
- [24] M. A. Lestari, M. Tabrani, and S. Ayumida, "SISTEM INFORMASI PENGOLAHAN DATA ADMINISTRASI KEPENDUDUKAN PADA KANTOR DESA PUCUNG KARAWANG," vol. 13, no. 3, p. 8, 2018.
- [25] "Apa itu ERD? Kenali Jenis, Komponen dan Tools yang Digunakan," *Sekawan Media / Software House & System Integrator Indonesia*, Jan. 04, 2021. <https://www.sekawanmedia.co.id/blog/apa-itu-erd/> (accessed Dec. 28, 2021).